AD_____

GRANT NO: DAMD17-94-J-4137

TITLE: Predicting Time-to-Relapse in Breast Cancer Using Neural Networks

PRINCIPAL INVESTIGATOR(S): Dr. Jonathan D. Buckley

CONTRACTING ORGANIZATION: University of Southern California
Los Angeles, California 90033

REPORT DATE: October 1995

TYPE OF REPORT: Annual

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for public release;
distribution unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1995	Annual 15 Sep 94 - 14 Sep 95

4. TITLE AND SUBTITLE
Predicting Time-to-Relapse in Breast Cancer Using Neural Networks

5. FUNDING NUMBERS
DAMD17-94-J-4137

6. AUTHOR(S)
Dr. Jonathan D. Buckley

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Southern California
Los Angeles, California 90033

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Neural networks (NN) have become established as powerful tools for complex pattern recognition problems. One application which appears well suited to NN methods is the identification of prognostic groups, to be used for treatment planning. For many cancer studies of cancer cell biology have added many factors of potential prognostic value, but the way in which these interact with known factors is generally not well studied. The potential of NNs to model these data in a non-linear fashion has only begun to be explored. NNs are not part of standard statistical packages, making them relatively inaccessible to many statisticians. More importantly, current NN methods cannot accommodate censored outcome variables.

This proposal is for application of several proposed methods for applying NNs to censored-data to the problem of predicting time-to-relapse for breast cancer patients. The methods will be evaluation in comparison to each other, and also to more conventional approaches such as Cox regression and recursive partitioning. The data on which the analyses will be conducted comes from the database of the NSABP.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Neural Networks, Survival, Relapse, Breast Cancer	15
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

FOREWORD

Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the US Army.

____ Where copyrighted material is quoted, permission has been obtained to use such material.

____ Where material from documents designated for limited distribution is quoted, permission has been obtained to use the material.

DLB Citations of commercial organizations and trade names in this report do not constitute an official Department of Army endorsement or approval of the products or services of these organizations.

____ In conducting research using animals, the investigator(s) adhered to the "Guide for the Care and Use of Laboratory Animals," prepared by the Committee on Care and Use of Laboratory Animals of the Institute of Laboratory Resources, National Research Council (NIH Publication No. 86-23, Revised 1985).

____ For the protection of human subjects, the investigator(s) adhered to policies of applicable Federal Law 45 CFR 46.

____ In conducting research utilizing recombinant DNA technology, the investigator(s) adhered to current guidelines promulgated by the National Institutes of Health.

____ In the conduct of research utilizing recombinant DNA, the investigator(s) adhered to the NIH Guidelines for Research Involving Recombinant DNA Molecules.

____ In the conduct of research involving hazardous organisms, the investigator(s) adhered to the CDC-NIH Guide for Biosafety in Microbiological and Biomedical Laboratories.

J.D. Buckley 8/23/95.

PI - Signature Date

TABLE OF CONTENTS

INTRODUCTION

Neural networks - a general description

NNs have been so-named because they mimic, in some respects, the structure and function of neurons in the brain. A NN consists of layers of nodes (analogous to neurons) linked by inter-connections (axons/dendrites), together with rules that specify how the output of each node is determined by input values from all nodes at the level below. A layered architecture of neurons in the brain can be used to provide progressively more abstract representation of input stimuli as the information is filtered through successive layers; NNs attempt to reproduce this effect, although most networks are limited in practice to three or four layers in total.

The lowest level of nodes in a NN is used to represent the input values, and the node or nodes in the highest level provide output from the NN. Since each node receives input from all nodes at the level below, generally combined as a weighted sum, the number of interconnections (and thus the number of weights) can be very large. Determining values for these weights *a priori* in order to obtained desired outputs for given inputs is clearly impractical for all but the most trivial networks. Useful NNs are made possible by the application of a learning algorithm that iteratively modifies the weights to minimize an "error function". The error function summarizes the differences between the actual output of the NN and the desired (or "true") output (Rumelhart, 1986).

The role of prognostic grouping and outcome prediction in clinical medicine

Establishing the prognosis for a patient may assist that patient in making **choices about treatment** and/or lifestyle changes. For breast cancer, determining the prognosis of a patient has become an essential first step to determining treatment: patients with a poor prognosis (at high risk of relapse or recurrence and/or with substantial residual disease) will generally be placed on the most intensive treatments while those with a good prognosis may be spared the acute toxicity and risks of long term effects associated with aggressive treatment.

The traditional methods used to identify prognostic variables are logistic regression for categorical outcomes, such as death/non-response/remission, or Cox regression (Cox, 1972) for survival-type outcomes. These multivariate methods generally combine the explanatory variables in a single linear expression; more complex relationships between the explanatory and outcome variables can be modelled using stratification and interaction terms but incorporation of such terms tend to be limited. However, a complex "prognostic syndrome" that involves several variables in a non-linear fashion would almost certainly escape attention in a traditional Cox regression analysis.

In recent years, the largely clinical data that have been used to separate prognostic groups have come to be supplemented to an increasing degree by laboratory data. New analytic methods can provide information on such things as specific mutations, gene amplification, gross chromosomal abnormalities such as translocations, deletions, ploidy changes etc., presence or absence or cell surface markers including antigens and receptor proteins, and immunological parameters. Not

unexpectedly, many of these biological characteristics correlate with outcome, but all too commonly new factors are reported without analysis of the extent to which they provide **independent** prognostic information, nor any guidance as to their use in conjunction with other factors in clinical decision making.

Use of neural networks to predict time to relapse in breast cancer

NNs have been used successfully to predict categorical clinical outcomes but there is no established method for dealing with potentially censored output values.

Ravdin et al (1992) published the first report of the use of NNs for clinical prediction with a survival-type outcome. This analysis attempted to relate six prognostic factors (tumor hormone status, DNA index, S-phase determination, tumor size, number of axillary nodes involved, and patient age) to time to relapse for women with node-positive breast cancer. A rather complex ad hoc method was used to adapt conventional NN programs to handle the censored data: (1) Selected input variables were log transformed, and normalized to lie within -1 to 1, (2) The database was split into a training set, evaluation set and validation set, (3) Time intervals (from the Kaplan-Meier curve) that corresponded to estimated rates of 0.90, 0.80, ... 0.10 were determined, (3) Each patient-record was split into m_i patient-time records (for patient i), where the time from study entry to time of analysis (the maximum follow-up) was T_i years and m_i of the time intervals come before T_i, (4) The NN was constructed with one output node (dead/alive) and a time variable (1, 2, up to T_i) as an input value. Patients that died before T_i were represented as dead in all patient-time records for intervals after their time of death, (5) To correct for bias due to non-uniform follow-up, the number of patient-time records corresponding to each interval was adjusted (by random elimination of records) to ensure that the ratio of records with "alive" status to those with "dead" status matched the observed Kaplan-Meier rates for the study group, (6) The output prediction was interpreted as a measure of relapse risk, and used to create risk subsets, (7) NN and Cox regression were compared in their ability to define groups with different Kaplan-Meier disease-free curves in an independent validation dataset, (8).

This approach was effective in defining prognostic groups: generally, the NN defined high and low risk subsets as efficiently as Cox regression and in some respects performed better. For instance, although having ten or more positive nodes was identified as a poor prognostic factor (32% relapse rate at 3 years), the NN placed only 54% of such patients in the high risk tertile; 40% were in the mid tertile and 6% were assigned to the lowest tertile. When the actual outcomes of the women with 10+ nodes were compared to all other women, within each predicted-risk tertile, relapse rates were very similar, indicating that the NN had correctly identified subgroups of apparent high risk (according to conventional methods) that belonged in lower risk groups.

Aims of this project

The aims stated in the original application were:

1. To develop a program designed to apply neural network methods to the analysis of clinical data. Development will involve two stages:

a. Software development of a neural network (NN) program, based on established methods.
b. Extension of the NN program to handle censored data.

2. To integrate this program with existing software, in order to:

   a. Provide the neural network program access to a wide range of database management/data transformation functions.
   b. Provide a single package that will perform traditional analyses of clinical data (e.g. Cox regression) and neural network modelling.

3. To evaluate alternative methods for identification of prognostic factors. The methods will be Cox regression (including recursive partitioning), censored linear regression, and four different neural network methods.

The projected timeline was that 1 and 2 would be completed within the first year, and that we would concentrate on aim 3 in year 2.

BODY

The following report details the progress made towards completion of aims 1 and 2 above.

Development of a general NN program

The neural network program has been developed as a procedure (PROC NEURAL) within the statistical package Epilog Plus, in order to benefit from the broad range of data management features of this program and to facilitate comaprisons with more conventional methods (Buckley, 1993). PROC NEURAL has been developed to have the basic features (not specifically related to analysis of survival-type data):

Basic structure: Feed-forward neural network, with up to four layers, up to 50 input nodes and 50 output nodes. Logistic transfer function. Dynamic changes to network structure through switching on or off of nodes.

Training: Back-propagation of errors (calculated as sum of square of prediction error). Logicon Projection offered as an option for weight initialization (see below). Weight updating following each record, or batched (e.g. after each 'run' through the training dataset). User-specifiable learning coefficient and momentum term, with the option to change these learning parameters after a preset number of runs - repeated such adjustments are allowed.

Epilog commands: Initial set-up determined by Epilog-style commands (see Appendix 3). Training pauses after a preset number of runs, or when the user 'breaks'. At this point, the commands can be modified, using the Epilog Plus command editor - if the changes do not alter the network structure, training continues from the previous point, otherwise the weights are re-initialized.

Training display. A graphical interface has been developed for PROC NEURAL, and during

training the user can view the following (Fig 2):

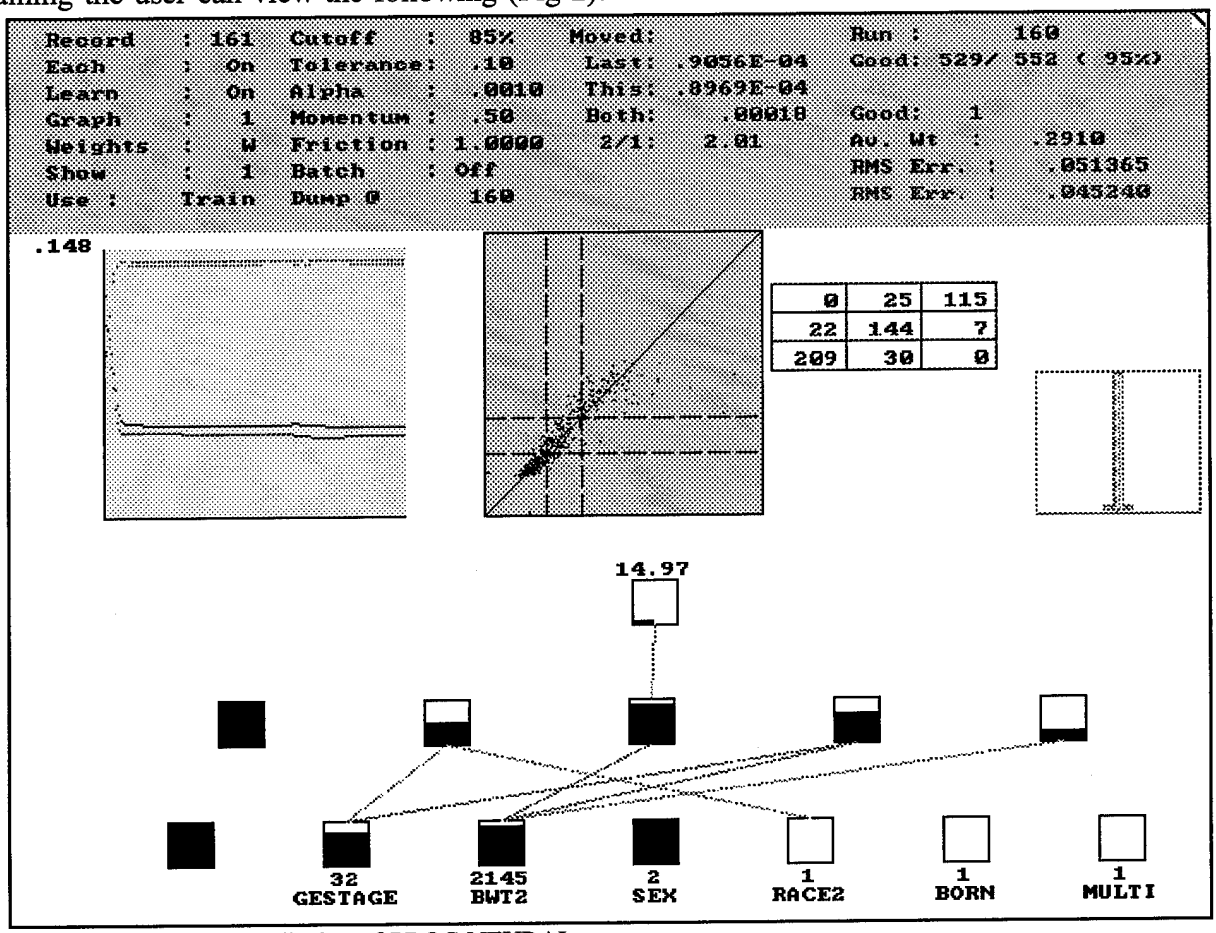

Figure 2. The training display of PROC NEURAL

An **upper data panel** that shows such things as the current learning parameters values, the run number, the distance the weight matrix has moved on the last two updates, and the root mean square (RMS) prediction error (for both the training dataset and a test dataset if it is available).

Plots that show (1) RMS error by run number (for training and test data separately) and percent of predictions that are within a user-specified tolerance value of the true value (also for training and test datasets), by run number; a scatter plot of predicted vs. actual value for a specified output node; a table of predicted vs. actual, based on the scatterplot and user-specified cutpoints; a histogram of the NN weights, which is useful for monitoring training progress.

A schematic of the network. Each node is shown, and for a selected training record, the values (and variable names) for each input and output node (for this record) are shown. The magnitude of the output from each node is indicated by a 'thermometer' that fills from 0% to 100% of the node interior. The record selected to be shown in this way can be fixed, or may change every time the display is refreshed. The display can be refreshed after every n runs (user specifiable). Selected interconnections between nodes are shown:

only those with a weight (or optionally, a weight times node value) that exceed a threshold are shown, with negative and positive connections distinguished by color.

Network analysis. When training halts, the user may continue using modified Epilog commands (see above), or may use single key-strokes to change model parameters before continuing. Other key-strokes are provided to allow the user to explore the current network and examine its properties. Options available include (1) Examining the effect on network performance of removing a selected node, (2) Display, on the network schematic, of all weights into and out of a selected node (Fig 3), (3) Display on the schematic of the inputs to a selected node (weight times outputs from previous level), (4) Distribution of outputs from a node for all cases in the training dataset, (5) Indication on the schematic of which input to a selected node made the largest contribution - expressed as the percent of training records for which each input was most influential, and (6) Indication on the schematic of the average contribution from each input to that node.

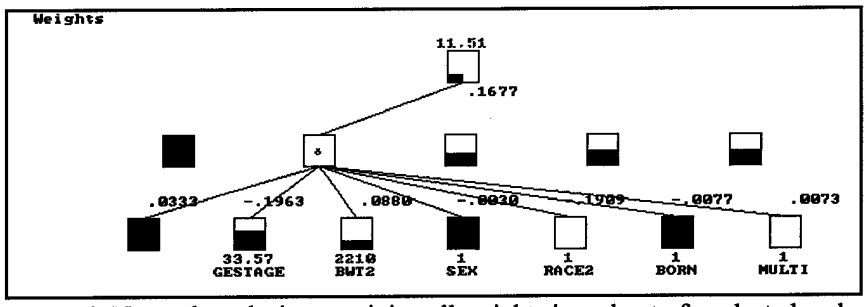

Figure 3. Network analysis: examining all weights in and out of a selected node.

Other features. The network periodically writes the weights to a disk file. This can be at set intervals (number of runs), or when the RMS error on the test dataset begins to increase (suggesting that the network is beginning to overfit the training dataset). The weights can subsequently be read back, to pick up training or other activities involving the network from any point at which a weight 'dump' occurred.

A single key-stroke requests that the network's **predicted outputs** be written into the database. These predictions are then available to all other Epilog procedures, for example to determine means, medians, distributions, and to graphically display using PROC GRAPH.

At any time the test and training dataset can be interchanged. This allows the user to examine network function with respect to test data as well as training data.

Extensions to the NN program to handle censored data

We proposed to develop the algorithms necessary for incorporation of several censored-data methods into the NN program. Progress in this direction is as follows:

Undefined node method.

The simplest approach is to represent the outcome as a series of indicator variables corresponding to periods of follow-up. The interval in which a death occurred is represented by a 1, and all other intervals (output nodes) take value 0. Intervals after a censoring event are considered to have an undefined value. In practice this means that this node does not contribute to the

prediction error - essentially, it has no influence in the error that is used (through back-propagation) to adjust the weights. This method represents a relatively straight-forward modification to a 'standard' NN, and has been implemented.

Buckley-James method.

The Buckley-James (expected value) method required more extensive changes to the NN code in order to implement. In this method, NN predictions are compared to the actual values on each run and the differences (residuals) used to calculate a Kaplan-Meier-type curve. Based on the residual distribution (as reflected in the Kaplan-Meier curve) it is possible to estimate the expected survival for any person who was censored. The Buckley-James approach, as it was described for linear regression (Buckley and James, 1979) and as it is generalized to the NN setting, is to determine the expected survival for all censored individuals (based on the current weight matrix), and to substitute the expected value for the censored value when determining and back-propagating the error. This method has been incorporated within PROC NEURAL.

Modified error function.

This approach involves a change to the error function so that the error calculated for censored observations penalizes under-prediction much more severely that over-prediction. This seems reasonable, since predictions less than the observed time are clearly in error; those above the observed time may or may not be. The error for uncensored observations remains equal to the squared difference between actual and predicted survival. We proposed in the Phase I application to use the function:

$$E_i = \alpha \exp\{\beta(O_i - P_i)\}$$

where E_i,O_i and P_i are the error, observed (actual), and predicted values for output node i, respectively, and α and β are parameters that influence the relative balance between censored and uncensored error terms and the rate at which the error term increases as the actual time exceeds the prediction (Katz S, 1993). The back-propagation algorithm for this error function differs only marginally from that used for a sum-of-squares error function: a term equal to the derivative of the error function with respect to a output (which is simply the difference between the actual and predicted output values for the sum-of-squares error function) is replaced by the equivalent derivative for the above function. This extension has been added to PROC NEURAL as an option.

Method of Ravdin

This method does not require any specific programming, since it was developed to use existing NN software. However, the incorporation of our NN software within a more general purpose software package will make the Ravdin method much simpler to apply.

Other additions relevant to this project

PROC COX and PROC CENREG

As part of the strategy of providing a powerful suite of routines, in one package, that could be used for clinical prediction using censored data, we have modified PROC COX (Cox regression) and PROC CENREG (censored linear regression) to generate predictions on a case-by-case basis and to write these back into the database.

PROC PARTITION

We proposed to compare NNs to the method of recursive partitiioning, and to this end have developed a recursive partitioning procedure (PROC PARTITION). PROC PARTITION is designed for censored data outcomes, and has the following features: (1) It allows up to 200 prognostic variables, which may be of continuous, binary or nominal type, (2) At each partitioning stage, the variable that provides the 'best' division of an existing partition is used to create a new partition, (3) The criterion for deciding on the best division may be based on ratios of observed to expected events, on degree of separation of the survival curves, or on the logrank statistic (p-value), (4) Partitioning ceases when this criterion (for the best division) does not exceed a prespecified threshold value, (5) For nominal variables the program tests all possible combinations of categories when searching for the best division, (6) For continuous variables, the program tests all possible cutpoints when searching for the best division, (6) Once partitioning is complete, the program will (optionally) examine all pairwise combinations of partitions to determine if any are sufficiently similar (based on O/E, separation or logrank statistic) to combine ("pruning"), (7) After two partitions are combined, all pairs are again examined, until no further combinations are possible. This PROC has been written and is currently undergoing testing.

Learning methods **(parameter estimation)**

Training a NN is simply an iterative method for estimating the model parameters (i.e. interconnection weights). The most common method - and the one we proposed to implement - is 'back-propagation' which is essentially a gradient descent approach. In practice, because of the large number of weights used in many NNs, convergence to an error minimum can be slow. Furthermore, this minimum may be a local rather than a global minimum. While we still feel that the back-propagation approach is extremely useful, the problems of long training time and local minima have led us to evaluate some alternative strategies for training of NNs.

Logicon projection.

This is an algorithm developed by scientists at Logicon Inc, Los Angeles (Wilensky G and Manukian N, 1992). It requires that the user specify a 'prototype' individual to correspond with each hidden node. A method of N-dimensional projection is used to calculate initial weights into these hidden nodes so that that node fires maximally for the prototype individual. Starting the NN with such weights, instead of randomly assigned one, can reduce training time by one to two orders of magnitude. Even if no care is taken to select appropriate prototypes, and they are drawn at random from the training database, training times can be substantially reduced.

We have implemented the Logicon Projection algorithm within PROC NEURAL.

Genetic algorithms.

A radically different approach to weight optimization may be used to try to avoid getting caught in a local minimum. With the so-called 'genetic algorithm' the weights are represented conceptually as genes on a chromosome. Instead of a single NN, a whole 'population' of networks with the same structure are created. The performance of each is evaluated, and the best (smallest error) are selected for 'mating', while the worst are removed (die). The weight-chromosome for each offspring is derived from the parent chromosomes through a process analogous to meiotic recombination with or without point mutations. Through many generations, with only the fittest being allowed to pass on their weight-chromosomes to new individuals, network performance improves. As a result of the discontinuous nature of the recombination process the weight matrix makes jumps in the parameter space that potentially avoid the trap of a local minimum and hopefully allows for exploration of the entire space for a global minimum (Narayanan and Lucas, 1993).

Our evaluation of genetic algorithms indicate that their implementation within PROC NEURAL is entirely feasible.

Applying the NN to simulated data.

The most useful database for initial evaluation of network performance is one in which the relationship between the input and output variables in known. For this reason we have relied heavily on data generated by PROC DIST of Epilog Plus. While it is not possible to present results from all such simulated databases, a single example is presented in detail below.

The database included 1000 training records and 1000 testing records with four binary input covariates (A, B, C and D), with the probability of a 1 being 0.05, 0.10, 0.25 and 0.50 respectively. Cases were assigned to a Low, Intermediate or High risk group, based on their covariate values; survival time was randomly generated from a negative exponential distribution, with hazards of 0.005, 0.01 and 0.02 respectively for the three risk groups. The relationship of covariate values to risk group was designed to provide a test of the NN's ability to detect complex interactions between covariates. Specifically those with A=1, or (C=1 and D=1) or (B=1 and C=1) were assigned to the High risk group; those with C=1 or (B=1 and D=1) were assigned to the Low group, while the remainder were Intermediate risk. The censor time was drawn from a uniform(0,365) distribution.

This database has been particularly valuable as a testing ground for NN under development. For example, a NN was created with a single hidden layer of three nodes, trained using the Buckley-James method and used to make predictions of outcome on the 1000 training and 1000 test cases. We used scatterplots and Cox goodness of fit to assess performance. Scatterplots were possible since the 'true' (uncensored) time was known for each case.

Cox regression was used to fit the four covariates (A - D), the NN prediction (P), a model with A,B,C,D and P, to determine whether P contained useful prognostic information not provided by the covariates (as main effects) - see Table 2. Note, these evaluations were based on the test group only.

Table 2. Cox regression goodness-of-fit chi-squares

	Model	Chi-square	D.O.F	p-value
1	A,B,C,D	44.00	4	<0.0001
2	P	90.86	1	<0.0001
3	A,B,C,D, plus P ..... vs. Model 1	106.54 62.54	5 1	<0.0001 <0.0001
4	A,B,C,D and all two-way interactions	105.96	10	<0.0001
5	A,B,C,D and all two-way interactions, plus P ..... vs. Model 4	118.68 12.72	11 1	<0.0001 <0.0001
6	Risk groups (Low, Intermediate, High)	111.54	2	<0.0001
7	Risk groups, plus P ..... vs. Model 6	111.84 0.30	4 1	<0.0001 N.S.

From these results we conclude that the NN prediction was able to substantially improve the Cox regression model fit when added to the four covariates, and even improved on a Cox model that included all two-way covariate interactions. As expected, it did not improve on a model in which the 'true' risk group assignments were represented. It is of interest that the NN prediction did improve on the risk-group model within the training dataset, illustrating the potential for NN models to over-fit the training data and underscoring the need for a separate testing dataset.

Since the 4 covariate variables can take only 16 possible combination of values, it is possible to examine the NN prediction for all input combinations. In addition we calculated the median of the fitted Cox distribution for models with A-D, and A-D, plus P. Comparison of predicted vs. median actual (uncensored) data are shown in Table 3.

Since most (84%) of the cases have one of four input patterns (0000,0001,0010 or 0011), it might be expected that the NN would train preferentially to fit these combinations, perhaps at the expense of the less frequently encountered combinations. In the above table, the predictions are compared to the actual mean values, with the exception of the NN prediction which is compared to both the mean and median. In theory, the NN should be predicting the mean, but it can be seen that its prediction was an underestimate in 9 instances and an overestimate in 5. More importantly, for the four most common patterns it underestimated by 12, 24 and 73 days, and was 1 day over in the remaining pattern. The reason for this tendency to underestimate is not clear, but is the subject of current research. Comparison of the NN prediction to the median indicated that the prediction tended to exceed the median.

The weighted absolute difference was calculated as a measure of the overall accuracy, allowing for the different frequency of covariate combinations. Based on this measure, the NN prediction was less useful than the Cox median value (based on A-D), but the Cox median based on A-D plus the NN prediction (P) was far superior to both.

Table 3. Comparison of predictive ability for all possible combinations of input variables

Group		Cases		Actual values		Difference (Predicted - Actual)			
ABCD	Risk	p	N	Mean	Median	NN vs. mean	NN vs. median	Cox (A-D)	Cox (P,A-D)
0000	Int	0.32	290	99	68	-12	+19	+6	+1
0001	Int	0.32	350	114	76	-24	+14	-7	-3
0010	Low	0.11	102	196	116	-73	+7	-41	+4
0011	High	0.11	100	64	51	+1	+14	+19	-2
0100	Int	0.04	35	108	97	-19	-8	-8	-23
0101	Low	0.04	49	183	132	-59	-8	-53	+5
0110	High	0.01	11	36	14	+16	+38	+76	+28
0111	Int	0.01	10	114	67	-66	-19	+13	-28
1000	High	0.016	14	46	16	+2	+32	+16	+16
1001	High	0.016	23	42	17	+3	+28	+10	+15
1010	High	0.006	4	51	41	+10	+20	-9	+1
1011	High	0.006	5	40	44	-9	-13	-17	-22
1100	High	0.002	2	26	9	-20	-3	+29	+6
1101	High	0.002	4	62	31	0	+31	+3	+16
1110	High	0.001	1	29	29	-29	-29	+10	-15
1111	High	0.001	0	-	-	-	-	-	-
Weighted absolute difference (days)						23.8	14.1	14.6	4.3

Breast cancer dataset

In preparation for year 2, which will involve analysis of data on breast cancer patients to comapre methods and establish a prognostic coding scheme, we have been collaborating with investigators at the NSABP data center. Initially this was with Dr Redmond, and after she stepped down as principal ststistician, with John Bryant. Dr. Van Tornout visited the NSABP data center to discuss the project in detail and learn as much as possible about the datasets that they have available. Subsequently we were fortunate that Dr. Bryant was able stop in Los Angeles en route to a meeting in San Diego, and we were able to follow up on the earlier meeting. As a result of those meetings it was decided that data from the B15 protocol would

be most appropriate for use on this project, at least initially. We expect to have a copy of the relevent data before year 2 begins.

CONCLUSIONS

The software-development phase of this project has gone smoothly. The tools needed to carry out a comprehensive comparison of methods for predicting time-to-relapse (specially NN methods) are either in place or will be very shortly. Thus we feel this project is right on schedule, and expect to complete the overall goals within the 2 year framework.

REFERENCES

Buckley JD, James IR. Linear regression with censored data. Biometrika 66:429-36, 1979.

Buckley JD. Epilog Plus manual. Epicenter Software, 1993.

Cox DR. Regression models and life tables. JR Stat Soc B 34:187-220, 1972.

Narayanan MN and Lucas SB. A genetic algorithm to improve a neural network to predict a patient's response to warfarin. Meth Inform Med 32:55-8, 1993.

Ravdin PM, Clark GM, Hilsenbeck SG, Owens MA, Vendely P, Pandian MR, McGuire WL. Demonstration that breast cancer recurrence can be predicted by neural network analysis. Breast Canc. Res. and Treat. 21:47-53, 1992.

Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back propagating. Nature 323:533-6, 1986.

Wilensky G and Manukian N. The projection neural network. Intl Joint Conf on Neural networks Vol II:358-67, 1992.